UNOFFICIAL COMMUNICATION FOR EXAMINER REVIEW ONLY – PLEASE DO NOT ENTER

LAW OFFICES
AMIN, TUROCY & CALVIN, LLP
57TH FLOOR, KEY TOWER
127 PUBLIC SQUARE
CLEVELAND, OHIO 44114
DIRECT TELEPHONE: 216-535-7950
MAIN TELEPHONE: 216-696-8730
FACSIMILE: 216-696-8731
EMAIL: EPERRY@THEPATENTATTORNEYS.COM

FACSIMILE TRANSMISSION

Date:	**February 6, 2009**
TO:	**Marina Lee – United States Patent and Trademark Office**
TEL NO.:	**571-270-2648**
FROM:	**Evan Perry**

In re patent application of:

Applicant(s):	Anson Horton et al.	Examiner:	Marina Lee
Serial No:	10/809,147	Art Unit:	2192
Filing Date:	March 25, 2004		

Title: PROXY OBJECTS FOR DISPLAY

TOTAL NUMBER OF PAGES (INCLUDING THIS PAGE): **2**

UNOFFICIAL COMMUNICATION FOR EXAMINER REVIEW ONLY – PLEASE DO NOT ENTER

Dear Examiner Lee:

Thank you for the opportunity to discuss the above captioned application on Tuesday, February 10, 2009 at 11:00am. The following agenda is provided to indicate the proposed topics of discussion.

I. Discussion of Rejection of Claims under 35 U.S.C. §103(a)

A. Discuss deficiencies of the cited art vis a vis the independent claims

1. The claimed subject matter relates to an attributed debugging system that enables a developer to associate a specific view of an object for examining in a debugger. The specific view includes only object information necessary for the developer to debug the object. The specific view is generated by a display proxy associated with the object. A debugger substitutes the display proxy for the object when examining the object
2. Dandoy relates to a user interface (UI) debugger. Dandoy discloses a debug agent that is combined with a software application. The debug agent collects execution data relating to graphical user interfaces during runtime and proves collected data to a UI debugger. For instance, the debug agent correlates data objects (e.g., instantiated objects associated with a type class) to UI elements in the interface. The debug agent can further monitor the user interface for events. The debug agent provides execution data to a UI debugger or other debugger upon a user request. Moreover, the debug agent can alter properties of UI object during debugging.
3. Bates et al. relates to debugging software code wherein the debugger includes additional descriptive material associated with variables beyond the values stored by the variables. In one

embodiment, comments associated with a variable are displayed in a debugger in connection with the variable.

4. The cited art fails to teach or suggest a display proxy that is examined by an expression evaluator in place of an object. Dandoy discloses a UI debugger that obtains properties of UI objects from instantiated UI objects themselves. Bates et al. discloses examining variables themselves and not a proxy.

Attached, please find a listing of claims with respect to the above-referenced matter. Thank you for your time and consideration.

Best regards,

-Evan

UNOFFICIAL COMMUNICATION FOR EXAMINER REVIEW ONLY – PLEASE DO NOT ENTER

Proposed Claim Amendments

1. (Currently amended) A computer-implemented attributed debugging system comprising:

a debugger that facilitates debugging of a computer software application, the debugger obtains values of one or more properties of an object of the computer software application;

an expression evaluator, associated with the debugger, that examines ~~determines values of one or more properties of an object of the computer software application being debugged, the expressions evaluator determines the values of the one or more properties of the object based upon~~ a display proxy in place of the object, the display proxy is implemented as a private nested class of the object, the display proxy ~~provides~~ is configured to expose a subset of the one or more properties of the object, the subset excludes implementation-specific properties of the object; and ~~relevant features of the object and conceals implementation specifics of the object; and~~

a variable display component that presents, to a developer, ~~the determined~~ values associated with the subset of the one or more properties of the object exposed by the display proxy. ~~to a developer.~~

2. (Previously presented) The system of claim 1, the expression evaluator evaluates an expression associated with the object to determine the values, the expression is implemented in a particular programming language.

3. (Previously presented) The system of claim 2, the programming language comprising at least one of C#, J# or Visual Basic.Net.

4. (Previously presented) The system of claim 2, further comprising a plurality of expression evaluators, wherein each expression evaluator is associated with a different programming language.

5. (Previously presented) The system of claim 1, the object comprises a class that includes at least one of a property or a method.

6. (Previously presented) The system of claim 1, the expression evaluator creates an instance of the display proxy associated with the object of the computer software application.

7. (Cancelled)

8. (Previously presented) The system of claim 1, the display proxy has access to private implementation specifics of the object.

9. (Previously presented) The system of claim 1, further comprising an attribute cache directory that stores an attribute associated with the display proxy, the expression evaluator employs the stored attribute to determine the values of the one or more properties of the object.

10. (Previously presented) The system of claim 1, further comprising an editing component that facilitates modifying a value associated with the object.

11. (Previously presented) The system of claim 1, the variable display employs at least one attribute associated with the object that provides a format to display the determined values of one or more properties of the object.

12. (Previously presented) The system of claim 11, the attribute specifies the property is displayed.

13. (Previously presented) The system of claim 12, the attribute employs an enumeration to specify the format of the display.

14. (Previously presented) The system of claim 13, the enumeration includes one enumeration value that indicates the property should not be displayed to the developer.

15. (Previously presented) The system of claim 13, the enumeration includes one enumeration value that indicates a hierarchical property is expanded by default.

16. (Previously presented) The system of claim 13, the enumeration includes one enumeration value that indicates a hierarchical property is not expanded by default.

17. (Previously presented) The system of claim 13, the enumeration includes one enumeration value that indicates a hierarchical property itself is not displayed and members of the hierarchical property are displayed.

18. (Previously presented) The system of claim 11, the attribute specifies what is displayed for a class.

19. (Previously presented) The system of claim 18, the attribute includes an argument that comprises a string that is displayed in a value column for an instance of the class.

20. (Previously presented) The system of claim 18, the argument is associated with a property of the class.

21. (Currently amended) A method facilitating attributed debugging comprising:

receiving a request to examine details of one or more properties of an object in a computer software application being debugged;

determining whether a display proxy attribute exists for the object, the display proxy is implemented as a private nested class of the object such that the display proxy within the definition of the object, the display proxy provides relevant properties regarding a state of the object and conceals properties related to implementation of the object;

creating a display proxy for the object in accordance with the display proxy attribute; and

examining the display proxy in place of the object to determine debug information related to the object.

22. (Previously presented) The method of claim 21 further comprising providing the debug information determined by the display proxy to a developer in response to the request to examine the object.

23. (Original) A computer readable medium having stored thereon computer executable instructions for carrying out the method of claim 21.

24-25. (Cancelled)

26. (Currently amended) An attributed debugging system comprising:

means for receiving a request to debug an object in a computer software application;

means for determining whether a display proxy attribute exists in association with the object, the display proxy is implemented as a private nested class of the object such that the display proxy within the definition of the object, the display proxy provides relevant properties regarding a state of the object and conceals properties related to implementation of the object;

means for creating an instance of the display proxy;

means for examining the display proxy in place of the object, means for examining the display proxy comprise means for obtaining values of the provided relevant properties; and

means for providing debug information in response to the request to debug the object, the debug information is based at least in part on the display proxy and includes at least the obtained values of the provided relevant properties.

27. (Currently amended) A computer-implemented attributed debugging system, comprising:

a debugger that facilitates debugging a computer software application;

an expression evaluator associated with the debugger that examines at least one object of the computer software application, the expression evaluator determines debug information comprising states of the at least one object, the states include values for at least one property of the at least one object,

the expression evaluator inspects the at least one object to verify if each of the at least one object includes a display proxy defined as a private nested class of the object, the display proxy provides relevant properties of an associated object and conceals properties related to implementation of the object, the expression evaluator examines the display proxy in place of the object to determine debug information that includes values for the relevant properties of the at least one object;

an attribute cache directory that retains instances of one or more display proxies, the expression evaluator queries the attribute cache directory for instances of display proxies associated with the at least one object, the expression evaluator creates an instance if not retained in the attribute cache directory; and

a variable display component that presents the debug information to a developer, the debug information includes values of relevant properties of the at least one object determined from examination of display proxies of the at least one object.